

The ooTest Framework

(Testing the ooRexx Interpreter)

Version 1.0.0 Edition
Draft - SVN Rev 7054
June 23, 2009



Mark Miesfeld

The ooTest Framework: (Testing the ooRexx Interpreter)

by

Mark Miesfeld

Version 1.0.0 Edition

Published June 23, 2009

Copyright © 2008, 2009 Rexx Language Association. All rights reserved.

This program and the accompanying materials are made available under the terms of the [Common Public License Version 1.0](#).

Before using this information and the product it supports, be sure to read the general information under [Notices](#).

Thanks to Julian Choy for the ooRexx logo design.

Table of Contents

About This Book	vii
1. Related Information	vii
2. How to Read the Syntax Diagrams	vii
3. Getting Help	viii
3.1. The Rexx Language Association Mailing List	ix
3.2. The Open Object Rexx SourceForge Site	ix
3.3. comp.lang.rexx Newsgroup	x
1. Overview of ooTest	1
1.1. Install ooTest	1
1.2. History, Foundation	2
1.3. Testing Concepts Used in ooTest	2
2. testOORexx.rex the Automated Test Driver	3
2.1. Using testOORexx.rex	3
2.2. Modes of Operation	5
2.3. Understanding the Output	5
2.4. Static Options	5
2.4.1. The -a "allTestTypes" option	5
2.4.2. The -b "buildFirst" option	6
2.4.3. The -B "forceBuild" option	6
2.4.4. The -d "defaultTestTypes" option	6
2.4.5. The -D "define" option	6
2.4.6. The -e "testContainerExt" option	6
2.4.7. The -f "singleFile" option	6
2.4.8. The -F "fileList" option	6
2.4.9. The -I "testTypeIncludes" option	6
2.5. Dynamic Options	6
2.6. The Options File	7
2.7. The .testOpts Directory	7
2.8. Some Debug Help	7
2.9. External Binaries	7
3. The ooRexxUnit Framework	9
3.1. TestCase Class	9
3.1.1. ooTestType (Class)	9
3.1.2. new (Class)	9
3.2. Assert Class	9
4. The ooTest Framework	11
4.1. General Concepts	11
4.1.1. Test Types	11
4.2. ooTestCase Class	12
4.2.1. ooTestType (Class)	12
4.2.2. new (Class)	13
4.3. ooTestTypes Class	13
4.3.1. ooTestType (Class)	13
4.3.2. new (Class)	13

- 5. Writing Tests 15**
- A. Notices 17**
 - A.1. Trademarks 17
 - A.2. Source Code For This Document 18
- B. Common Public License Version 1.0 19**
 - B.1. Definitions 19
 - B.2. Grant of Rights 19
 - B.3. Requirements 20
 - B.4. Commercial Distribution 20
 - B.5. No Warranty 21
 - B.6. Disclaimer of Liability 21
 - B.7. General 22
- Index 23**

List of Tables

3-1. ooTestCase Summary	9
3-2. ooTestCase Summary	9
4-1. ooTestCase Summary	12
4-2. ooTestCase Summary	13

About This Book

ooTest in the broadest sense represents all the tools and components used to test ooRexx and all of the tests written to date. Minus the written tests, ooTest is a testing framework that facilitates the testing of the ooRexx interpreter. It sits on top of the ooRexxUnit framework which is a generic testing framework that anyone can use to test their Rexx applications.

The audience for this book is primarily those interested in testing the ooRexx interpreter. Either by executing the test suite or by writing test cases. In the process of explaining how to write test cases for the ooRexx interpreter, the book covers the ooRexxUnit and ooTest frameworks. For this reason portions of the book may be of interest to persons wishing to use ooRexxUnit to test their own Rexx applications.

This book does not provide information on how to program using ooRexx. The assumption is that the reader is already familiar with the interpreter and, to some degree, the use of objects in ooRexx. The book provides a reference to `testOORexx.rex`, the ooRexxUnit framework, the ooTest framework, and information on how to write tests of the ooRexx interpreter package using the ooTest framework.

The book is roughly organized this way:

- An [overview](#) of ooTest.
- A chapter documenting [testOORexx.rex](#) which is currently the one and only test driver program.
- Chapter(s) documenting the [ooRexxUnit](#) framework.
- Chapter(s) documenting the [ooTest](#) framework.
- Chapter(s) giving some [guidance](#) in writing test cases using the ooTest framework.

1. Related Information

See also: *Open Object Rexx: Reference*

2. How to Read the Syntax Diagrams

Throughout this book, syntax is described using the structure defined below.

- Read the syntax diagrams from left to right, from top to bottom, following the path of the line.

The `>>---` symbol indicates the beginning of a statement.

The `--->` symbol indicates that the statement syntax is continued on the next line.

The `>---` symbol indicates that a statement is continued from the previous line.

The `---><` symbol indicates the end of a statement.

Diagrams of syntactical units other than complete statements start with the `>---` symbol and end with the `--->` symbol.

- Required items appear on the horizontal line (the main path).

```
>>-STATEMENT--required_item-----><
```

- Optional items appear below the main path.

```
>>-STATEMENT--+-+-----><
                +-optional_item-+
```

- If you can choose from two or more items, they appear vertically, in a stack. If you must choose one of the items, one item of the stack appears on the main path.

```
>>-STATEMENT--+-required_choice1-+-----><
                +-required_choice2-+
```

- If choosing one of the items is optional, the entire stack appears below the main path.

```
>>-STATEMENT--+-+-----><
                +-optional_choice1-+
                +-optional_choice2-+
```

- If one of the items is the default, it appears above the main path and the remaining choices are shown below.

```
                +-default_choice--+
>>-STATEMENT--+-+-----><
                +-optional_choice-+
                +-optional_choice-+
```

- An arrow returning to the left above the main line indicates an item that can be repeated.

```
                +-----+
                v           |
>>-STATEMENT----repeatabl_item+-----><
```

A repeat arrow above a stack indicates that you can repeat the items in the stack.

- A set of vertical bars around an item indicates that the item is a fragment, a part of the syntax diagram that appears in greater detail below the main diagram.

```
>>-STATEMENT--| fragment |-----><
```

fragment:

```
|--expansion_provides_greater_detail-----|
```

- Keywords appear in uppercase (for example, PARM1). They must be spelled exactly as shown but you can type them in upper, lower, or mixed case. Variables appear in all lowercase letters (for example, parm x). They represent user-supplied names or values.
- If punctuation marks, parentheses, arithmetic operators, or such symbols are shown, you must enter them as part of the syntax.

The following example shows how the syntax is described:

```
                +-,-----+
                v           |
>>-MAX(----number+---)-----><
```

3. Getting Help

The Open Object Rexx Project has a number of methods to obtain help for ooRexx. These methods, in no particular order of preference, are listed below.

3.1. The Rexx Language Association Mailing List

The *Rexx Language Association* (<http://www.rexxla.org/>) maintains a mailing list for its members. This mailing list is only available to RexxLA members thus you will need to join RexxLA in order to get on the list. The dues for RexxLA membership are small and are charged on a yearly basis. For details on joining RexxLA please refer to the *RexxLA Home Page* (<http://rexxla.org/>) or the *RexxLA Membership Application* (<http://www.rexxla.org/rexxla/join.html>) page.

3.2. The Open Object Rexx SourceForge Site

The Open Object Rexx Project (<http://www.oorexx.org/>) utilizes *SourceForge* (<http://sourceforge.net/>) to house the *ooRexx Project* (<http://sourceforge.net/projects/oorexx>) source repositories, mailing lists and other project features. Here is a list of some of the most useful facilities.

The ooRexx Forums

The ooRexx project maintains a set of forums that anyone may contribute to or monitor. They are located on the *ooRexx Forums* (http://sourceforge.net/forum/?group_id=119701) page. There are currently three forums available: Help, Developers and Open Discussion. In addition, you can monitor the forums via email.

The Developer Mailing List

You can subscribe to the oorexx-devel mailing list at *ooRexx Mailing List Subscriptions* (http://sourceforge.net/mail/?group_id=119701) page. This list is for discussing ooRexx project development activities and future interpreter enhancements. It also supports a historical archive of past messages.

The Users Mailing List

You can subscribe to the oorexx-users mailing list at *ooRexx Mailing List Subscriptions* (http://sourceforge.net/mail/?group_id=119701) page. This list is for discussing using ooRexx. It also supports a historical archive of past messages.

The Announcements Mailing List

You can subscribe to the oorexx-announce mailing list at *ooRexx Mailing List Subscriptions* (http://sourceforge.net/mail/?group_id=119701) page. This list is only used to announce significant ooRexx project events.

The Bug Mailing List

You can subscribe to the oorexx-bugs mailing list at *ooRexx Mailing List Subscriptions* (http://sourceforge.net/mail/?group_id=119701) page. This list is only used for monitoring changes to the ooRexx bug tracking system.

Bug Reports

You can create a bug report at *ooRexx Bug Report* (http://sourceforge.net/tracker/?group_id=119701&atid=684730) page. Please try to provide as much information in the bug report as possible so that the developers can determine the problem as quickly as possible. Sample programs that can reproduce your problem will make it easier to debug reported problems.

Request For Enhancement

You can suggest ooRexx features at the *ooRexx Feature Requests* (http://sourceforge.net/tracker/?group_id=119701&atid=684733) page.

Patch Reports

If you create an enhancement patch for ooRexx please post the patch using the *ooRexx Patch Report* (http://sourceforge.net/tracker/?group_id=119701&atid=684732) page. Please provide as much information in the patch report as possible so that the developers can evaluate the enhancement as quickly as possible.

Please do not post bug patches here, instead you should open a bug report and attach the patch to it.

3.3. comp.lang.rexx Newsgroup

The comp.lang.rexx (news:comp.lang.rexx) newsgroup is a good place to obtain help from many individuals within the Rexx community. You can obtain help on Open Object Rexx or on any number of other Rexx interpreters and tools.

Chapter 1. Overview of ooTest

The primary purpose of ooTest is to thoroughly test the ooRexx interpreter and its distribution packages. In general, to thoroughly test software involves automating as much as possible of the testing. The ooTest framework provides the foundation for automatically executing tests. It currently contains little or no components that would aid in automating the writing of tests.

Note: This document is written for the 4.0.0 and later versions of ooTest. It makes no effort to cover the 3.2.0 version of ooTest, or explain any differences between the 3.2.0 and 4.0.0 versions. There are differences.

ooTest also contains all the tests written to date to that the ooRexx team uses to test ooRexx. This makes ooTest itself more than just a framework. It is the sum of the tests themselves, the framework to write the tests, and any other components used in the overall process of testing the ooRexx package.

This document serves as a reference to the different components of ooTest. It explains how to execute the tests, interpret the results, and gives some guidance on writing new tests. There is very little involved in obtaining and setting up ooTest. Indeed there is very little set up involved in ooTest. It is merely a matter of placing the ooTest files on the system used to test.

1.1. Install ooTest

Getting and installing ooTest is extremely easy. There is no set up involved, you merely place the ooTest directory structure on somewhere convenient on your system. The best method of obtaining the ooTest directory structure is to use a Subversion client to check out the ooTest repository. Subversion is widely available, easy to install, and easy to use. Directions for obtaining Subversion have been posted numerous times to the lists related to ooRexx. There is a wealth of information on this subject on the Internet, simply use Google.

Once you have a Subversion client, the following command will check out ooTest and place its directory structure in the subdirectory named ooTest.4.0.0. The subdirectory can be named anything you wish.

```
C:\>svn co https://oorexx.svn.sourceforge.net/svnroot/oorexx/test/trunk ooTest.4.0.0
```

The command is exactly the same on any operating system.

The other way to obtain ooTest is to download one of the ooTest snapshots from the ooRexx project on SourceForge. The drawback to this method is that the latest snapshot may not be current and you can not use the Subversion tools to keep your copy of ooTest up-to-date. To download a snapshot go to this link on SourceForge: *SourceForge.net: Open Object Rexx Files* (https://sourceforge.net/project/showfiles.php?group_id=119701&package_id=251951) and download the appropriate file. Then either untar or unzip the file wherever is convenient. For example:

```
C:\>unzip ooTest-4.0.0-snapshot04_windows.i386.zip
```

Once the directory structure is placed on your system, you are ready to go. Use the driver program, [testOORexx.rexx](#) to execute some tests. The tests should run and print out some statistics. This will ensure ooTest is working. The rest of this book goes into detail on all aspects of ooTest.

1.2. History, Foundation

ooTest started out as ooRexxUnit. ooRexxUnit is a *generic* testing framework designed to serve as a unit testing framework for anyone to use to test their Rexx programs. It is patterned after jUnit and originally strove to emulate jUnit as closely as possible. As a publicly released piece of software it was constrained by typical needs, such as providing backward compatibility, limiting changes that require the user to make changes to their existing testing efforts, etc..

Over time it became apparent that the constraints of ooRexxUnit were a hinderance to the ooRexx testing effort. Therefore, the ooTest framework was put in place. ooTest sits on top of ooRexxUnit, but it is not designed to be a generic testing framework, does not guarantee backward compatibility, and does not strive to closely emulate jUnit. The idea behind this was to leave ooRexxUnit as a generic testing framework for any one to use, and focus the ooTest framework on specifically testing ooRexx.

However, because of its history, ooTest is conceptually similar to jUnit. If you understand jUnit concepts, you will have little trouble understanding ooTest. The main difference is that jUnit is primarily a unit testing framework, while ooTest is primarily an automation framework, that heavily supports unit testing.

From the perspective of writing test cases, ooTest is very jUnit-like. The terminology of ooTest is very similar to that of jUnit. ooTest uses test cases, test case classes, test suites, test results, test runners, and asserts, just as jUnit does. ooTest uses the same design patterns that jUnit uses. Hardly surprising as ooRexxUnit was modeled after jUnit. jUnit has generated many tutorial and primer types of documentation on the Internet. Google 'jUnit tutorial' or 'jUnit primer' and browse some of the simpler ones. This will quickly give you a grasp of ooTest test cases.

1.3. Testing Concepts Used in ooTest

ooTest is a way of writing defined repeatable tests, collecting and storing the tests, and executing all or some of those tests, now and in the future. With ooTest, all tests are to test the ooRexx interpreter and its distribution package. The purpose of a test is to demonstrate that some facet of the interpreter works, both now and at the next point in time the test is executed. To do this ooTest uses objects and the following loosely discusses these objects and how they fit together.

Test Case

The test case is the basic concept of ooTest. It is what the collection of defined repeatable tests is built upon.

Assertion

x.

Chapter 2. testOORexx.rex the Automated Test Driver

The execution of all the tests in the test suite is currently driven by one Rexx program, `testOORexx.rex`. This program has an extensive set of options that allow the execution of all the tests, a single test, or subsets of the tests. As is all of `ooTest`, the program is platform independent and runs on all platforms where `ooRexx` is available.

When run, `testOORexx` sets up the environment, gathers up the tests to be executed, executes them, and prints out the statistics for the test run.

There is nothing preventing the use of other driver programs with the tests for the interpreter. It is merely that no one has written one. In particular, a GUI test driver would be nice and relatively easy to produce. One approach would be for the GUI driver to present the interface and then call `testOORexx` under the covers to do the actual work.

2.1. Using testOORexx.rex

`testOORexx.rex` is a command line program. It is designed to, and should be, run from a console window. Starting `testOORexx.rex` is the same as starting any Rexx program. On Windows, with a default install of `ooRexx` where `.rex` has a file association with the interpreter, you can simply type `testOORexx`. For example:

```
C:\ooTest>testOORexx
```

On an unix-like system, `testOORexx` would typically be started like this:

```
[miesfeld@falcon]# ./testOORexx.rex
```

Of course `testOORexx` can always be started this way:

```
C:\ooTest>rexx testOORexx.rex
```

```
[miesfeld@falcon]# rexx testOORexx.rex
```

Usually you would start `testOORexx` in the root directory of `ooTest`. I.e., in the directory that contains `testOORexx.rex`. However, `testOORexx.rex` is designed to run from any location on the local machine's file system. The main purpose of this is to allow a developer to execute the test suite from within their build directory. But, as the following example illustrates, `testOORexx` does not need to be run from the `ooTest` directory or a build directory, it can be run from anywhere on the system:

```
D:\TravelDrive>c:\ooTest\testOORexx.rex -F SimpleTests Searching for test containers..
Executing automated test suite..
```

```
ooTest Framework - Automated Test of the ooRexx Interpreter
```

```
Interpreter:      REXX-ooRexx_4.0.0(MT) 6.03 21 Jun 2009
Addressing Mode: 64
ooRexxUnit:      2.0.0_3.2.0    ooTest: 1.0.0_4.0.0

Tests ran:        9
Assertions:       9
Failures:         0
Errors:           0
Skipped files:   0

File search:      00:00:00.000000
Suite construction: 00:00:00.000000
Test execution:   00:00:00.000000
Total time:      00:00:01.000000
```

```
D:\TravelDrive>
```

Although running testOORexx will correctly set up the PATH to execute the tests, some people may prefer to change the path in the environment. This will allow running testOORexx from anywhere without having to type in the complete path name of testOORexx, such as C:\ooTest\testOORexx. In addition, on Windows testOORexx is able to completely set up the environment for executing all the tests, but on unix-like systems the library path can not be set. On the unix-like systems, the library path **must be** set prior to executing any test requiring one of the [external binaries](#). Two small utility scripts, setTestEnv.bat and setTestEnv.sh are provided to handle these chores. On Windows:

```
C:\ooTest>setTestEnv
```

On a unix-like system be sure to source the utility script:

```
[miesfeld@falcon]# ./setTestEnv.sh
Setting env for Linux
[miesfeld@falcon]#
```

Another reason to use the utility scripts is that this allows a TODO test group ADD LINK to be run as a stand alone Rexx program from its current directory. This is sometimes useful when first developing a set of test cases in a test group. For example:

```
C:\ooTest\ooRexx\base\class>rexx Array.testGroup
```

```
Interpreter:      REXX-ooRexx_4.0.0(MT) 6.03 21 Jun 2009
Addressing Mode: 64
ooRexxUnit:      2.0.0_3.2.0    ooTest: 1.0.0_4.0.0

Tests ran:        101
Assertions:       8007
Failures:         0
Errors:           0
Skipped files:   0
```

```
Test execution:      00:00:00.469000
```

```
C:\ooTest\ooRexx\base\class>
```

Executing a test group as a stand alone program is not possible unless the ooTest root directory is in the path because the interpreter will not be able to find the ooTest.frm file:

```
C:\ooTest\ooRexx\base\class>rexx Array.testGroup
  54 *-* ::requires 'ooTest.frm' -- load the ooRexxUnit classes
Error 43 running C:\ooTest\ooRexx\base\class\Array.testGroup line 54: Routine not found
Error 43.901: Could not find routine "ooTest.frm" for ::REQUIRES
```

```
C:\ooTest\ooRexx\base\class>
```

Note: As a reminder, on unix-like systems, sourcing `setTestEnv.sh` is a requirement when tests using the TODO external binaries ADD LINK will be executed. Unless of course the library path is manually set by some other means.

2.2. Modes of Operation

To Be Written

2.3. Understanding the Output

To Be Written

2.4. Static Options

The term *static* is used here for lack of a better one. These are the *known* or *implemented* options as opposed to the *dynamic* options formed by using the `-D` option.

Each of the known options has a short form and a long name form. The long name for the option is used when specifying the option in the [options file](#). The long name can also be used on the command line in combination with the `-D` option if it is easier to remember the long name.

The following sections discuss each of the known options individually.

2.4.1. The -a "allTestTypes" option

This option is a boolean option, specified by true or false. The default is false.

2.4.2. The -b "buildFirst" option

To Be Written

2.4.3. The -B "forceBuild" option

To Be Written

2.4.4. The -d "defaultTestTypes" option

To Be Written

2.4.5. The -D "define" option

To Be Written

2.4.6. The -e "testContainerExt" option

To Be Written

2.4.7. The -f "singleFile" option

To Be Written

2.4.8. The -F "fileList" option

To Be Written

2.4.9. The -I "testTypeIncludes" option

To Be Written

2.5. Dynamic Options

Options need not be known to the `testOORexx` program. Options can be specified dynamically by defining the name of the option and its value on the command line, or in the [options file](#). These options are handled generically by `testOORexx`.

2.6. The Options File

To Be Written

2.7. The `.testOpts` Directory

To Be Written

2.8. Some Debug Help

To Be Written

2.9. External Binaries

To Be Written

Chapter 3. The ooRexxUnit Framework

x

3.1. TestCase Class

A *TestCase* class

The following table summarizes the important information for a *TestCase* class, including all methods and attributes relevant to writing test cases. Most of these methods are inherited from *TestCase*:

Table 3-1. ooTestCase Summary

Item	...Description
Something	some link

3.1.1. ooTestType (Class)

```
>>--ooTestType-----<<
```

```
>>--ooTestType=-----<<
```

To Be Written

3.1.2. new (Class)

```
>>--new-----<<
```

To Be Written

3.2. Assert Class

An *Assert* class

The following table summarizes the important information for an *Assert* class, including all methods and attributes relevant to writing test cases. Most of these methods are inherited from *TestCase*:

Table 3-2. ooTestCase Summary

Item	...Description
Something	some link

Chapter 4. The ooTest Framework

The ooTest framework requires the [ooRexxUnit framework](#), that is it uses a `::requires` directive for `OOREXXUNIT.CLS`. It extends the functionality of ooRexxUnit in ways that are not necessarily intended to be generic, but rather are only intended to serve some specific need in testing ooRexx. Most of the main classes in ooTest are direct subclasses of a comparable class in ooRexxUnit, adding only minor functionality to the ooRexxUnit class.

This section of the book serves as a reference to the public classes, routines, and features of the ooTest framework. It starts with a general discussion of where and how the framework differs from a generic unit testing framework and the objects in the framework that implement these differences. The rest of the material is the reference documentation for the classes and routines that the framework provides.

The reference material is intended to be most useful to people writing test cases for ooRexx. Please **note** how this is structured. Many of the classes and methods of the framework are used to relieve the test case writer of much, or most, of the burden of implementation. The test case writer, for the most part, only needs the documentation for a few classes and usually only for a few methods of those classes. The rest of the classes and methods need only be understood by the maintainers of the overall ooRexx testing project.

However, it makes little sense to document part of a class here, the part used by test case writers, and the rest of the class in some other place, say a *maintainers* manual. So this strategy is used: the primary classes a test case writer would use are documented first. Within a class documentation, the methods and attributes most useful in writing test cases are also documented first. The rest of the classes and methods will be marked with the keyword: *internal*. This doesn't mean these classes or methods are only for internal use. It is merely meant to indicate to the reader, who is mostly interested in writing test cases, that she can probably ignore the classes and methods.

4.1. General Concepts

Description

4.1.1. Test Types

Test types are implemented through the [ooTestTypes](#) and [ooTestCase](#) classes.

Unlike junit, which only does unit tests, ooTest is designed to automate all the testing for the ooRexx distribution package. There is of course more to software testing than unit tests. There are stress tests, performance tests, acceptance tests (does the distribution package install correctly, do all the samples at least run,) GUI tests, etc..

The value of an automated unit testing framework comes from developers running the unit tests often. It has been suggested that preferably the unit tests should be run after every compile. But, if ooTest is going to be the framework used to run stress tests, acceptance tests, etc., the time to execute the test suite starts growing exponentially. In addition stress or performance tests typically use large amounts of system resources. If the developer's machine doesn't have the required horse power for a stress test, what then? If an ooRexx user wants to help by writing and executing test cases, but doesn't have access to a smtp server needed to run the socket test cases, how does he proceed?

Clearly, if ooTest is to serve as an all-purpose test framework, there needs to be some way of easily running subsets of the tests that are applicable at the time. The developer needs to be able to easily run only the unit tests. The user trying to contribute to the project, but is lacking a smtp server, needs some way to run the test suite, but exclude the socket tests.

ooTest handles this by using the concept of *test types*. In junit there is only one type of test, the unit test. In ooTest, each group of test cases in an `ooTestCase` subclass belongs to one of any number of defined test types. The `ooTestCase` class has a class attribute that defines the test type the class contains. By default the test cases in an `ooTestCase` subclass are the unit test type.

When writing a `ooTestCase` subclass, the test type for the subclass can be over-ridden. This is done for example in the *Native_api* tests. The class `init()` method is over-ridden as in this example:

```
::class "CONVERSION.testGroup" subclass ooTestCase public

::method init class
  forward class (super) continue

  -- Over-ride the default test type
  self~ooTestType = .ooTestTypes~NATIVE_API_TEST
```

4.2. ooTestCase Class

An *ooTestCase* class is a test case class where methods of the class define individual test cases.

In order to make it easy to construct automated tests with large numbers of test cases, a convention is followed: Each method of an `ooTestCase` class that starts with *test* is considered an individual test case.

Each `ooTestCase` has a class attribute defining the *test type* of the individual test cases the class contains. All test cases in a specific `ooTestCase` class are the same test type. The class `new()` method is intended to be over-ridden by test case writers to assign an appropriate test type when needed.

The following table summarizes the important information for an `ooTestCase` class, including all methods and attributes relevant to writing test cases. Most of these methods are inherited from *TestCase*:

Table 4-1. ooTestCase Summary

Item	...Description
Subclasses	TestCase
Inherits	ooTestTypes
Inherits (indirectly through TestCase)	Assert
Class Attributes	
ooTestType (Class Attribute)	ooTestType
Class Methods	
new (Class method)	new

4.2.1. ooTestType (Class)

```
>>--ooTestType-----<<
```

```
>>--ooTestType=-----<<
```

To Be Written

4.2.2. new (Class)

```
>>--new-----<<
```

To Be Written

4.3. ooTestTypes Class

An *ooTestTypes* class

The following table summarizes the important information for an ooTestCase class, including all methods and attributes relevant to writing test cases. Most of these methods are inherited from *TestCase*:

Table 4-2. ooTestCase Summary

Item	...Description
Something	some link

4.3.1. ooTestType (Class)

```
>>--ooTestType-----<<
```

```
>>--ooTestType=-----<<
```

To Be Written

4.3.2. new (Class)

```
>>--new-----<<
```

To Be Written

Chapter 5. Writing Tests

x

Appendix A. Notices

Any reference to a non-open source product, program, or service is not intended to state or imply that only non-open source product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any Rexx Language Association (RexxLA) intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-open source product, program, or service.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-open source products was obtained from the suppliers of those products, their published announcements or other publicly available sources. RexxLA has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-RexxLA packages. Questions on the capabilities of non-RexxLA packages should be addressed to the suppliers of those products.

All statements regarding RexxLA's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

A.1. Trademarks

Open Object Rexx™ and ooRexx™ are trademarks of the Rexx Language Association.

The following terms are trademarks of the IBM Corporation in the United States, other countries, or both:

1-2-3
AIX
IBM
Lotus
OS/2
S/390
VisualAge

AMD is a trademark of Advance Micro Devices, Inc.

Intel, Intel Inside (logos), MMX and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in

the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

A.2. Source Code For This Document

The source code for this document is available under the terms of the Common Public License v1.0 which accompanies this distribution and is available in the appendix *Common Public License Version 1.0*. The source code itself is available at

http://sourceforge.net/project/showfiles.php?group_id=119701.

The source code for this document is maintained in DocBook SGML/XML format.



Appendix B. Common Public License Version 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS COMMON PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

B.1. Definitions

"Contribution" means:

1. in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
2. in the case of each subsequent Contributor:
 - a. changes to the Program, and
 - b. additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents " mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

B.2. Grant of Rights

1. Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.
2. Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such

combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

3. Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.
4. Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

B.3. Requirements

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

1. it complies with the terms and conditions of this Agreement; and
2. its license agreement:
 - a. effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
 - b. effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
 - c. states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
 - d. states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

1. it must be made available under this Agreement; and
2. a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

B.4. Commercial Distribution

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

B.5. No Warranty

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

B.6. Disclaimer of Liability

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE

PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

B.7. General

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against a Contributor with respect to a patent applicable to software (including a cross-claim or counterclaim in a lawsuit), then any patent licenses granted by that Contributor to such Recipient under this Agreement shall terminate as of the date such litigation is filed. In addition, if Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. IBM is the initial Agreement Steward. IBM may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

Index

Symbols

.testOpts, [7](#)

A

Assert class, [9](#)

C

Common Public License, [19](#)
concepts
 ooTest framework, [11](#)
CPL, [19](#)

E

external binaries, [7](#)

I

install
 ooTest, [1](#)
installing ooTest, [1](#)

L

License, Common Public, [19](#)
License, Open Object Rexx, [19](#)

N

new
 ooTestCase class, [9](#), [13](#), [13](#)
Notices, [17](#)

O

ooRexx License, [19](#)
ooRexxUnit, [9](#)
ooTest, [15](#)
 foundation, [2](#)
 frameworks, [11](#)
 history, [2](#)
 overview, [1](#)
ooTest framework, [11](#)
ooTestCase class, [12](#)
ooTestType, [9](#), [13](#), [13](#)
 ooTestCase class, [9](#), [13](#), [13](#)
ooTestTypes class, [13](#)
Open Object Rexx License, [19](#)
options.ooTest, [7](#)

S

setTestEnv.bat, [4](#)
setTestEnv.sh, [4](#)

T

Test
 concepts, [2](#)
 execution, [3](#)
test types, [11](#)
TestCase class, [9](#)
testOORexx
 .testOpts directory, [7](#)
 debugging, [7](#)
 dynamic options, [7](#)
 external binaries, [7](#)
 modes, [5](#)
 options file, [7](#)
 output, [5](#)
 starting, [3](#)
 static options, [5](#)
testOORexx options
 allTestTypes, [6](#)
 buildFirst, [6](#)
 defaultTestTypes, [6](#)
 define, [6](#)
 fileList, [6](#)
 forceBuild, [6](#)

singleFile, [6](#)
testContainerExt, [6](#)
testTypeIncludes, [6](#)
testOORexx.rex, [3](#)